SUMMARY OF THE PRESENT INVENTION

In a first aspect, there is a method of enabling a wireless information device to automatically modify its behaviour, comprising the steps of:

(a) an end-user entering time sensitive information into a first application running on the device;

(b) a second application running on the device receiving data from the first application, the data relating to the time sensitive information, and the second application then automatically changing the behaviour of the device appropriately in dependence on the data;

in which the first application sends the data indirectly to the second application via an intermediary server.

Hence, the present invention deals with the very specific situation of an end-user entering time sensitive information into an application; this, for example, may be an entry (e.g. 'meeting', 'lunch with Bob', 'travelling', 'flying' etc) against specific times in an agenda or calendar application. Then, a different application on the device can utilise that information to modify the device behaviour appropriately. For example, say the 'meeting' in the calendar application is listed to last between 10am and 11am; then, during that hour, the telephone application in the device (that enables the telephone functions of the device to be controlled) could automatically be set to a suitable profile, such as a 'silent' profile so that the device does not ring on an incoming call, but instead only vibrates. The term 'application' should be expansively construed to cover any structure of software that performs one or more functions; it hence covers elements/portions of an operating system, utilities, client components, server components etc. Particularly, the second application could be OS system services, as opposed to an application which presents a unique interface to an end-user.

The first application sends the data indirectly to the second application via an intermediary server. The term 'server' is used in its normal, broad sense to mean a computer program that provides services to other computer programs. The server operates as an insulation layer, separating the first and second applications (the term 'insulation layer' is a term of art in design patterns). It allows there to be one or several

first applications (i.e. applications into which the user can input time sensitive information – such as (a) an agenda application into which the end-user can enter appointments, meetings, events etc. occurring at different defined times and (b) an alarm application into which the end-user can input a time he would like the device to sound an alarm). Further, the server enables any of these first applications to send data to any number of other applications that are to respond to the time sensitive information (or more specifically the data sent from the first application(s) that relate to this time sensitive information). Hence, a device might be in sleep mode, but the power management application be set to awake to normal mode when the alarm sounds: in this case, the first application is the alarm application and the second application is the power management application. The server can also ensure that the second application is running and to launch it if necessary.

The approach of the present invention therefore provides a structured and systematic way for a device to intelligently use time sensitive information that the end-user has manually input.

The end-user could, in relation to the agenda application entry (e.g. 'meeting between 10am and 11am) select from a menu list (i.e. any kind of user interface that enables the end-user to select different options) a label to apply to the entry, the label defining the type of behaviour change to be carried out by the second application. Hence, in relation to the 'meeting', 'lunch', 'travelling' or 'flying' etc. entries, the end-user could activate a pop-up menu of possible behaviour changes linked to that event. These might include the following selectable options:

(a) altering a telephone profile (e.g. silent, vibrate, loud, divert to voice mail etc.)
(b) altering the device ring tone
(c) altering the device user interface
(d) switching off telephone functionality
(e) switching off the device entirely
(f) switching the device to a power save mode

(g) switching off one or more items of communications hardware (e.g. GPS, IR, Bluetooth, wireless LAN modules).

Generally , the second application automatically changes the behaviour of the device appropriately in dependence on the data from the first application for a time period determined by that data, e.g. silent mode for 1 hour to coincide with the meeting duration.

Another example is that there could be a dialog near the weekdays dialog in the agenda application with pairs of times the end-user enters to put the phone into power save mode, weekday sleep times, Saturday sleep times, Sunday sleep times etc.

If a conflict arises between the behaviour change due to the data from the first application and a different behaviour change input directly by the end-user to the first or the second application, then the different behaviour change may prevail – different types of changes may be allocated different priorities. Hence, the end-user could place the device directly back into sleep mode as or shortly after the alarm is sounding, to continue with the example above.

In one implementation, dynamically ensuring event-precedence is deterministic - meaning all events must guarantee to resolve to correct behaviour. So if one event has higher priority than another, this must be true under all circumstances ('if and only if'), and if two events have the same priority they must resolve to the same result each time.

If a conflict arises between the behaviour change due to the data from the first application and a different behaviour change input directly to the first or the second application, then a conflict resolution component determines which behaviour change prevails. A component designed to resolve conflicts can have a pre-defined list of conflict situations and the appropriate conflict resolution decision it should make given those inputs. The pre-defined conflict list may contains derivation rules, i.e. allow algebraic substitution rather that a static scenario and are thus are dynamic and will not cause compatibility problems; this facilitates third parties subscribing and publishing with the system.

An override component could also be provided to determine if a behaviour change due to the data from the first application is inappropriate and to then override that behaviour change. Hence, a device may be automatically about to wake up from sleep mode and sound the alarm as described above; but the device may also have a location sensor that informs it that it is in an aircraft and hence must not activate. In this circumstance, the wake up behaviour change would be over-ridden. The intermediary server could provide the conflict resolution component and override component.

In addition, it is possible for the end-user to enter time sensitive information into the first application running on the device; and for a second application running on a second, different device to receive data directly or indirectly from the first application, the data relating to the time sensitive information, and the second application then automatically changing the behaviour of the second device appropriately in dependence on the data. In this way, for example, an alarm notification set into one device could be used to trigger an action within or by an application running on a completely different device; the two devices might be connected over a wireless link. Hence, it could trigger a web camera to activate and send images to a web site.

In a second aspect, there is a wireless information device programmed to automatically modify its behaviour, the device enabling:

(a) an end-user to enter time sensitive information into a first application running on the device;

(b) a second application running on the device to receive data directly or indirectly from the first application, the data relating to the time sensitive information, and the second application then automatically changing the behaviour of the device appropriately in dependence on the data;

in which the first application sends the data indirectly to the second application via an intermediary server.